IEEE Std 730-1998
(Revision of
IEEE Std 730-1989)

IEEE Standard for Software Quality Assurance Plans

Sponsor

Software Engineering Standards Committee of the IEEE Computer Society

Approved 25 June 1998

IEEE-SA Standards Board

Abstract: Uniform, minimum acceptable requirements for preparation and content of Software Quality Assurance Plans (SQAPs) are provided. This standard applies to the development and maintenance of critical software. For noncritical software, or for software already developed, a subset of the requirements of this standard may be applied.
Keywords: critical design review, preliminary design review, software configuration management plan, software design description, software quality assurance plan, software requirements review, software requirements specification, software verification and validation plan

---

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

ISBN 0-7381-0328-4

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes LaneP.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

(This introduction is not a part of IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.)

This standard assists in the preparation and content of Software Quality Assurance Plans and provides a standard against which such plans can be prepared and assessed. It is directed toward the development and maintenance of critical software, i.e., where failure could impact safety or cause large financial or social losses.

The readers of this document are referred to IEEE Std 730.1-1995 for recommended approaches to good software quality assurance practices in support of this standard. While IEEE Std 730.1-1995 specifically refers to IEEE Std 730-1984, almost all of its content applies directly to this revision.

The readers of this document are referred to Annex A for guidelines for using this document to meet the requirements of IEEE/EIA 12207.1-1997, IEEE/EIA Guide for Information Technology—Software life cycle processes—Life cycle data.

In this standard, firmware is considered to be software and is to be treated as such.

Footnotes are not part of the standard.

There are three groups to whom this standard applies: the user, the developer, and the public.

a) The user, who may be another element of the same organization developing the software, has a need for the product. Further, the user needs the product to meet the requirements identified in the specification. The user thus cannot afford a "hands-off" attitude toward the developer and rely solely on a test to be executed at the end of the software development time period. If the product should fail, not only does the same need still exist, but also a portion of the development time has been lost. Therefore, the user needs to obtain a reasonable degree of confidence that the product is in the process of acquiring required attributes during software development.

b) The developer needs an established standard against which to plan and to be measured. It is unreasonable to expect a complete reorientation from project to project. Not only is it not cost effective, but, unless there exists a stable framework on which to base changes, improvement cannot be made.

c) The public may be affected by the users' use of the product. These users include, for example, depositors at a bank or passengers using a reservation system. Users have requirements, such as legal rights, which preclude haphazard development of software. At some later date, the user and the developer may be required to show that they acted in a reasonable and prudent professional manner to ensure that required software attributes were acquired.

Participants

This standard was prepared by the Life Cycle Data Harmonization Working Group of the Software Engineering Standards Committee of the IEEE Computer Society. At the time this standard was approved, the working group consisted of the following members:

Leonard L. Tripp, *Chair*

Edward Byrne	Dennis Lawrence	Terry Rout
Paul R. Croll	David Maibor	Richard Schmidt
Perry DeWeese	Ray Milovanovic	Norman F. Schneidewind
Robin Fralick	James Moore	David Schultz
Marilyn Ginsberg-Finner	Timothy Niesen	Basil Sherlund
John Harauz	Dennis Rilling	Peter Voldner
Mark Henley		Ronald Wade

The following persons were on the balloting committee:

Syed Ali
Leo Beltracchi
H. Ronald Berlack
Richard E. Biehl
Michael A. Blackledge
Sandro Bologna
Juris Borzovs
Audrey C. Brewer
Kathleen L. Briggs
M. Scott Buck
Michael Caldwell
James E. Cardow
Jaya R. Carl
Enrico A. Carrara
Keith Chan
Antonio M. Cicu
Theo Clarke
Sylvain Clermont
Rosemary Coleman
Virgil Lee Cooper
W. W. Geoff Cozens
Paul R. Croll
Patricia W. Daggett
Gregory T. Daich
Geoffrey Darnton
Taz Daughtrey
Raymond Day
Bostjan K. Derganc
Perry R. DeWeese
Evelyn S. Dow
Charles Droz
Sherman Eagles
Robert G. Ebenau
Leo Egan
Richard L. Evans
William Eventoff
John W. Fendrich
Jay Forster
Kirby Fortenberry
Eva Freund
Richard C. Fries
Roger U. Fujii
Juan Garbajosa-Sopena
Barry L. Garner
Adel N. Ghannam
Marilyn Ginsberg-Finner
John Garth Glynn
Julio Gonzalez-Sanz
L. M. Gunther
David A. Gustafson
John Harauz
Herbert Hecht
William Hefley
Manfred Hein
Debra Herrmann
Umesh P. Hiriyannaiah
John W. Horch
Jerry Huller
Peter L. Hung
George Jackelen
Frank V. Jorgensen
Vladan V. Jovanovic
William S. Junk
George X. Kambic
Ron S. Kenett
Judith S. Kerner
Robert J. Kierzyk
Shaye Koenig
Thomas M. Kurihara
John B. Lane
J. Dennis Lawrence
Stanley H. Levinson
Fang Ching Lim
William M. Lively
John Lord
Stan Magee
David Maibor
Harold Mains
Mike McAndrew
Patrick D. McCray
Sue McGrath
Jerome W. Mersky
Bret Michael
Alan Miller
Celia H. Modell
James W. Moore
Pavol Navrat
Myrna L. Olson
Donald J. Ostrom
Indradeb P. Pal
Lalit M. Patnaik
Alex Polack
Peter T. Poon
Lawrence S. Przybylski
Kenneth R. Ptack
Annette D. Reilly
Dennis Rilling
Terence P. Rout
Andrew P. Sage
Stephen R. Schach
Norman Schneidewind
David J. Schultz
Lisa A. Selmon
Robert W. Shillato
David M. Siefert
Carl A. Singer
Nancy M. Smith
Alfred R. Sorkowitz
Donald W. Sova
Luca Spotorno
Fred J. Strauss
Christine Brown Strysik
Toru Takeshita
Richard H. Thayer
Douglas H. Thiele
Booker Thomas
Patricia Trellue
Mark-Rene Uchida
Theodore J. Urbanowicz
Glenn D. Venables
Andre Villas-Boas
Udo Voges
Dolores Wallace
William M. Walsh
John W. Walz
Camille S. White-Partain
Scott A. Whitmire
P. A. Wolfgang
Paul R. Work
Natalie C. Yopconka
Janusz Zalewski
Geraldine Zimmerman
Peter F. Zoll

When the IEEE-SA Standards Board approved this standard on 25 June 1998, it had the following membership:

Richard J. Holleman, *Chair*
Donald N. Heirman, *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon
James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry
L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Valerie E. Zelenty
IEEE Standards Project Editor

Contents

IEEE Standard for Software Quality Assurance Plans

1. Overview

1.1 Scope

The purpose of this standard is to provide uniform, minimum acceptable requirements for preparation and content of Software Quality Assurance Plans (SQAPs).

In considering adoption of this standard, regulatory bodies should be aware that specific application of this standard may already be covered by one or more IEEE standards documents relating to quality assurance, definitions, or other matters. It is not the purpose of this standard to supersede, revise, or amend existing standards directed to specific industries or applications.

This standard applies to the development and maintenance of critical software. For noncritical software, or for software already developed, a subset of the requirements of this standard may be applied.

The existence of this standard should not be construed to prohibit additional content in an SQAP. An assessment should be made for the specific software item to assure adequacy of coverage. Where this standard is invoked for an organization or project engaged in producing several software items, the applicability of the standard should be specified for each of the software items.

2. References

The standards listed below should be used for further information. In using these references, the latest revisions should be obtained. Compliance with this standard does not require nor imply compliance with any of those listed.

ASME NQA-1-1997, Quality Assurance Requirements for Nuclear Facility Applications.[1]

IEEE Std 7-4.3.2-1993, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.[2]

---

[1]ASME publications are available from the American Society of Mechanical Engineers, 22 Law Drive, Fairfield, NJ 07007, USA.
[2] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

IEEE Std 603-1998, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.[3]

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.

IEEE Std 828-1998, IEEE Standard for Software Configuration Management Plans.[4]

IEEE Std 829-1998, IEEE Standard for Software Test Documentation.[5]

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

IEEE Std 982.1-1988, IEEE Standard Dictionary of Measures to Produce Reliable Software.

IEEE Std 982.2-1988, IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software.

IEEE Std 1002-1987 (Reaff 1992), IEEE Standard Taxonomy of Software Engineering Standards.

IEEE Std 1008-1987 (Reaff 1993), IEEE Standard for Software Unit Testing.

IEEE Std 1012-1998, IEEE Standard for Software Verification and Validation.

IEEE Std 1012a-1998, IEEE Standard for Software Verification and Validation: Content Map to IEEE/EIA 12207.1-1997.[6]

IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions.[7]

IEEE Std 1028-1997, IEEE Standard for Software Reviews.

IEEE Std 1033-1985, IEEE Recommended Practice for Application of IEEE Std 828 to Nuclear Power Generating Stations.[8]

IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management.

IEEE P1058/D2.1, Draft Standard for Software Project Management Plans, dated 5 August 1998.[9]

IEEE Std 1058a-1998, IEEE Standard for Software Project Management Plans: Content Map to IEEE/EIA 12207.1-1997.[10]

IEEE Std 1063-1987 (Reaff 1993), IEEE Standard for Software User Documentation.

---

[3]As this standard goes to press, IEEE Std 603-1998; IEEE Std 828-1998; IEEE Std 829-1998; IEEE Std 1012a-1998; and IEEE Std 1016-1998 are approved but not yet published. The draft standards are, however, available from the IEEE. Anticipated publication date is Fall 1998. Contact the IEEE Standards Department at 1 (732) 562-3800 for status information.
[4]See Footnote 3.
[5]See Footnote 3.
[6]See Footnote 3.
[7]See Footnote 3.
[8]IEEE Std 1033-1985 has been withdrawn; however, copies can be obtained from Global Engineering, 15 Inverness Way East, Englewood, CO 80112-5704, USA, tel. (303) 792-2181.
[9]Upon approval of IEEE P1058 by the IEEE-SA Standards Board, this standard will be integrated with IEEE Std 1058a-1998 and published as IEEE Std 1058, 1998 Edition. Approval is expected 8 December 1998.
[10]As this standard goes to press, IEEE Std 1058a-1998 is approved but not yet published. The draft standard is, however, available from the IEEE. Anticipated publication date is December 1998. Contact the IEEE Standards Department at 1 (732) 562-3800 for status information. See Footnote 9.

3. Definitions and acronyms

3.1 Definitions

The definitions listed below establish meaning in the context of this standard. Other definitions can be found in IEEE Std 610.12-1990,[11] or the latest revision thereof. For the purpose of this standard, the term "software" includes firmware, documentation, data, and execution control statements (e.g., command files, job control language).

3.1.1 branch metric: The result of dividing the total number of modules in which every branch has been executed at least once by the total number of modules.[12]

3.1.2 critical software: Software whose failure would impact safety or cause large financial or social losses.

3.1.3 decision point metric: The result of dividing the total number of modules in which every decision point has had 1) all valid conditions, and 2) at least one invalid condition, correctly processed, divided by the total number of modules.[13]

3.1.4 domain metric: The result of dividing the total number of modules in which one valid sample and one invalid sample of every class of input data items (external messages, operator inputs, and local data) have been correctly processed, by the total number of modules.[14]

3.1.5 error message metric: The result of dividing the total number of error messages that have been formally demonstrated, by the total number of error messages.

3.1.6 quality assurance: A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements.

3.1.7 requirements demonstration metric: The result of dividing the total number of separately-identified requirements in the software requirements specification (SRS) that have been successfully demonstrated by the total number of separately-identified requirements in the SRS.

3.2 Acronyms

The following alphabetical contractions appear within the text of this standard:

CDR	critical design review
PDR	preliminary design review
SCMP	software configuration management plan
SCMPR	software configuration management plan review
SDD	software design description
SQA	software quality assurance
SQAP	software quality assurance plan
SRR	software requirements review
SRS	software requirements specification
SVV	software verification and validation plan
SVVPR	software verification and validation plan review
SVVR	software verification and validation report
UDR	user documentation review

[11] Information on references can be found in Clause 2.
[12] This definition assumes that the modules are essentially the same size.
[13] See Footnote 12.
[14] See Footnote 12.

4. Software Quality Assurance Plan

The Software Quality Assurance Plan shall include the sections listed below to be in compliance with this standard. The sections should be ordered in the described sequence. If the sections are not ordered in the described sequence, then a table shall be provided at the end of the SQAP that provides a cross-reference from the lowest numbered subsection of this standard to that portion of the SQAP where that material is provided. If there is no information pertinent to a section, the following shall appear below the section heading, "This section is not applicable to this plan," together with the appropriate reasons for the exclusion.

a) Purpose;
b) Reference documents;
c) Management;
d) Documentation;
e) Standards, practices, conventions, and metrics;
f) Reviews and audits;
g) Test;
h) Problem reporting and corrective action;
i) Tools, techniques, and methodologies;
j) Code control;
k) Media control;
l) Supplier control;
m) Records collection, maintenance, and retention;
n) Training;
o) Risk management.

Additional sections may be added as required.

Some of the material may appear in other documents. If so, then reference to these documents should be made in the body of the SQAP. In any case, the contents of each section of the plan shall be specified either directly or by reference to another document.

The SQAP shall be approved by the chief operating officer of each unit of the organization having responsibilities defined within this SQAP or their designated representatives.

Details for each section of the SQAP are described in 4.1 through 4.15 of this standard.[15]

4.1 Purpose (Section 1 of the SQAP)

This section shall delineate the specific purpose and scope of the particular SQAP. It shall list the name(s) of the software items covered by the SQAP and the intended use of the software. It shall state the portion of the software life cycle covered by the SQAP for each software item specified.

4.2 Reference documents (Section 2 of the SQAP)

This section shall provide a complete list of documents referenced elsewhere in the text of the SQAP.

[15]Guidance in the use of this standard can be found in IEEE Std 730.1-1995. For an expansion of the quality and equipment qualification requirements of IEEE Std 603-1998, to encompass software design, software implementation, and computer systems validation, see IEEE Std 7-4.3.2-1993.

4.3 Management (Section 3 of the SQAP)

This section shall describe organization, tasks, and responsibilities.[16]

4.3.1 Organization

This paragraph shall depict the organizational structure that influences and controls the quality of the software. This shall include a description of each major element of the organization together with the delegated responsibilities. Organizational dependence or independence of the elements responsible for SQA from those responsible for software development and use shall be clearly described or depicted.

4.3.2 Tasks

This paragraph shall describe

a) That portion of the software life cycle covered by the SQAP;
b) The tasks to be performed with special emphasis on software quality assurance activities; and
c) The relationships between these tasks and the planned major checkpoints.

The sequence of the tasks shall be indicated.

4.3.3 Responsibilities

This paragraph shall identify the specific organizational elements responsible for each task.

4.4 Documentation (Section 4 of the SQAP)

4.4.1 Purpose

This section shall perform the following functions:

a) Identify the documentation governing the development, verification and validation, use, and maintenance of the software.
b) State how the documents are to be checked for adequacy. This shall include the criteria and the identification of the review or audit by which the adequacy of each document shall be confirmed, with reference to Section 6 of the SQAP.

4.4.2 Minimum documentation requirements

To ensure that the implementation of the software satisfies requirements, the documentation in 4.4.2.1 through 4.4.2.6 is required as a minimum.

4.4.2.1 Software Requirements Specification (SRS)

The SRS shall clearly and precisely describe each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement shall be defined such that its achievement is capable of being objectively verified and validated by a prescribed method (e.g., inspection, analysis, demonstration, or test).[17]

[16]See IEEE Std 1002-1987, IEEE P1058/D2.1, and IEEE Std 1058a-1998.
[17]See IEEE Std 830-1998.

4.4.2.2 Software Design Description (SDD)

The SDD shall depict how the software will be structured to satisfy the requirements in the SRS. The SDD shall describe the components and subcomponents of the software design, including databases and internal interfaces. The SDD shall be prepared first as the Preliminary SDD (also referred to as the top-level SDD) and shall be subsequently expanded to produce the Detailed SDD.[18]

4.4.2.3 Software Verification and Validation Plan (SVVP)

The SVVP shall identify and describe the methods (e.g., inspection, analysis, demonstration, or test) to be used to[19]

a) Verify that
   1) The requirements in the SRS have been approved by an appropriate authority;
   2) The requirements in the SRS are implemented in the design expressed in the SDD; and
   3) The design expressed in the SDD is implemented in the code.
b) Validate that the code, when executed, complies with the requirements expressed in the SRS.

4.4.2.4 Software Verification and Validation Report (SVVR)

The SVVR shall describe the results of the execution of the SVVP.

4.4.2.5 User documentation

User documentation (e.g., manual, guide) shall specify and describe the required data and control inputs, input sequences, options, program limitations, and other activities or items necessary for successful execution of the software. All error messages shall be identified and corrective actions shall be described. A method of describing user-identified errors or problems to the developer or the owner of the software shall be described. (Embedded software that has no direct user interaction has no need for user documentation and is therefore exempted from this requirement.)[20]

4.4.2.6 Software Configuration Management Plan (SCMP)

The SCMP shall document methods to be used for identifying software items, controlling and implementing changes, and recording and reporting change implementation status.[21]

4.4.3 Other

Other documentation may include the following:

a) Software Development Plan;
b) Standards and Procedures Manual;
c) Software Project Management Plan;
d) Software Maintenance Manual.

[18]See IEEE Std 1016-1998.
[19]See IEEE Std 829-1998, IEEE Std 1008-1987, IEEE Std 1010-1998, and IEEE Std 1012a-1998.
[20]See IEEE Std 1063-1987.
[21]See IEEE Std 828-1998 and IEEE Std 1042-1987. See also IEEE Std 1033-1985.

4.5 Standards, practices, conventions, and metrics (Section 5 of the SQAP)

4.5.1 Purpose

This section shall

a) Identify the standards, practices, conventions, and metrics to be applied;
b) State how compliance with these items is to be monitored and assured.

4.5.2 Content

The subjects covered shall include the basic technical, design, and programming activities involved, such as documentation, variable and module naming, programming, inspection, and testing. As a minimum, the following information shall be provided:[22]

a) Documentation standards;
b) Logic structure standards;
c) Coding standards;
d) Commentary standards;
e) Testing standards and practices;
f) Selected software quality assurance product and process metrics such as
    1) Branch metric;
    2) Decision point metric;
    3) Domain metric;
    4) Error message metric;
    5) Requirements demonstration metric.

4.6 Reviews and audits (Section 6 of the SQAP)

4.6.1 Purpose

This section shall[23]

a) Define the technical and managerial reviews and audits to be conducted;
b) State how the reviews and audits are to be accomplished;
c) State what further actions are required and how they are to be implemented and verified.

4.6.2 Minimum requirements

As a minimum, the reviews and audits in 4.6.2.1 through 4.6.2.10 shall be conducted.

4.6.2.1 Software Requirements Review (SRR)

The SRR is held to ensure the adequacy of the requirements stated in the SRS.

4.6.2.2 Preliminary Design Review (PDR)

The PDR (also known as the top-level design review) is held to evaluate the technical adequacy of the preliminary design (also known as the top-level design) of the software as depicted in the preliminary software design description.

[22] See IEEE Std 990-1987, IEEE Std 982.1-1988, and IEEE Std 982.2-1988.
[23] See IEEE Std 1028-1997.

4.6.2.3 Critical Design Review (CDR)

The CDR (also known as detailed design review) is held to determine the acceptability of the detailed software designs as depicted in the detailed software design description in satisfying the requirements of the SRS.

4.6.2.4 Software Verification and Validation Plan Review (SVVPR)

The SVVPR is held to evaluate the adequacy and completeness of the verification and validation methods defined in the SVVP.

4.6.2.5 Functional audit

This audit is held prior to the software delivery to verify that all requirements specified in the SRS have been met.

4.6.2.6 Physical audit

This audit is held to verify that the software and its documentation are internally consistent and are ready for delivery.

4.6.2.7 In-process audits

In-process audits of a sample of the design are held to verify consistency of the design, including the following:

a) Code versus design documentation;
b) Interface specifications (hardware and software);
c) Design implementations versus functional requirements;
d) Functional requirements versus test descriptions.

4.6.2.8 Managerial reviews

Managerial reviews are held periodically to assess the execution of all of the actions and the items identified in the SQAP. These reviews shall be held by an organizational element independent of the unit being reviewed, or by a qualified third party. This review may require additional changes in the SQAP itself.

4.6.2.9 Software Configuration Management Plan Review (SCMPR)

The SCMPR is held to evaluate the adequacy and completeness of the configuration management methods defined in the SCMP.

4.6.2.10 Post-mortem review

This review is held at the conclusion of the project to assess the development activities implemented on that project and to provide recommendations for appropriate actions.

4.6.3 Other

Other reviews and audits may include the user documentation review (UDR). This review is held to evaluate the adequacy (e.g., completeness, clarity, correctness, and usability) of user documentation.

4.7 Test (Section 7 of the SQAP)

This section shall identify all the tests not included in the SVVP for the software covered by the SQAP and shall state the methods to be used.[24]

4.8 Problem reporting and corrective action (Section 8 of the SQAP)

This section shall

a) Describe the practices and procedures to be followed for reporting, tracking, and resolving problems identified in both software items and the software development and maintenance process;
b) State the specific organizational responsibilities concerned with their implementation.

4.9 Tools, techniques, and methodologies (Section 9 of the SQAP)

This section shall identify the special software tools, techniques, and methodologies that support SQA, state their purposes, and describe their use.

4.10 Code control (Section 10 of the SQAP)

This section shall define the methods and facilities used to maintain, store, secure, and document controlled versions of the identified software during all phases of the software life cycle. This may be implemented in conjunction with a computer program library. This may be provided as a part of the SCMP. If so, an appropriate reference shall be made thereto.

4.11 Media control (Section 11 of the SQAP)

This section shall state the methods and facilities to be used to

a) Identify the media for each computer product and the documentation required to store the media, including the copy and restore process; and
b) Protect computer program physical media from unauthorized access or inadvertent damage or degradation during all phases of the software life cycle.

This may be provided as a part of the SCMP. If so, an appropriate reference shall be made thereto.

4.12 Supplier control (Section 12 of the SQAP)

This section shall state the provisions for assuring that software provided by suppliers meets established requirements. In addition, this section shall state the methods that will be used to assure that the software supplier receives adequate and complete requirements. For previously developed software, this section shall state the methods to be used to assure the suitability of the product for use with the software items covered by the SQAP. For software that is to be developed, the supplier shall be required to prepare and implement an SQAP in accordance with this standard. This section shall also state the methods to be employed to assure that the developers comply with the requirements of this standard.

[24] See IEEE Std 829-1998 and IEEE Std 1008-1987.

4.13 Records collection, maintenance, and retention (Section 13 of the SQAP)

This section shall identify the SQA documentation to be retained; shall state the methods and facilities to be used to assemble, safeguard, and maintain this documentation; and shall designate the retention period.

4.14 Training (Section 14 of the SQAP)

This section shall identify the training activities necessary to meet the needs of the SQAP.

4.15 Risk management (Section 15 of the SQAP)

This section shall specify the methods and procedures employed to identify, assess, monitor, and control areas of risk arising during the portion of the software life cycle covered by the SQAP.

Annex A

Guidelines for compliance with IEEE/EIA 12207.1-1997

(Informative)

A.1 Overview

The Software Engineering Standards Committee (SESC) of the IEEE Computer Society has endorsed the policy of adopting international standards. In 1995, the international standard, ISO/IEC 12207, Information technology—Software life cycle processes, was completed. The standard establishes a common framework for software life cycle processes, with well-defined terminology, that can be referenced by the software industry.

In 1995 the SESC evaluated ISO/IEC 12207 and decided that the standard should be adopted and serve as the basis for life cycle processes within the IEEE Software Engineering Collection. The IEEE adaptation of ISO/IEC 12207 is IEEE/EIA 12207.0-1996. It contains ISO/IEC 12207 and the following additions: improved compliance approach, life cycle process objectives, life cycle data objectives, and errata.

The implementation of ISO/IEC 12207 within the IEEE also includes the following:

— IEEE/EIA 12207.1-1997, IEEE/EIA Guide for Information Technology—Software life cycle processes—Life cycle data;
— IEEE/EIA 12207.2-1997, IEEE/EIA Guide for Information Technology—Software life cycle processes—Implementation considerations; and
— Additions to 11 SESC standards (i.e., IEEE Stds 730, 828, 829, 830, 1012, 1016, 1058, 1062, 1219, 1233, 1362) to define the correlation between the data produced by existing SESC standards and the data produced by the application of IEEE/EIA 12207.1-1997.

NOTE — Although IEEE/EIA 12207.1-1997 is a guide, it also contains provisions for application as a standard with specific compliance requirements. This annex treats 12207.1-1997 as a standard.

A.1.1 Scope and purpose

Both this standard and IEEE/EIA 12207.1-1997 place requirements on an SQAP. The purpose of this annex is to explain the relationship between the two sets of requirements so that users producing documents intended to comply with both standards may do so.

A.2 Correlation

This clause explains the relationship between this standard and IEEE/EIA 12207.0-1996 in the following areas: terminology, process, and life cycle data.

A.2.1 Terminology correlation

The basic term for this standard is quality assurance. The definitions for the term from this standard and IEEE/EIA 12207.0-1996 are presented below:

quality assurance: A planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements. (IEEE Std 730-1998)

quality assurance: All the planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfill requirements for quality. (IEEE/EIA 12207.0-1996)

The two definitions are essentially the same, particularly if it is assumed that "technical requirements" plus quality expectations are the same as "requirements for quality." The remaining terminology in this standard reflects the assumptions prevalent in the 1980s about software engineering, but for the most part this does not affect the use or meaning of terms. There are two exceptions. The first is the use of the term "audit" in 4.6.2.7 in the phrase "in-process audit"; the word "audit" should be placed with "verification." The second is the concept of independence. This standard only requires organizational independence for managerial reviews whereas IEEE/EIA 12207.0-1996 requires "organizational freedom and authority from persons directly responsible for developing the software product or executing the process responsible for the product."

A.2.2 Process correlation

This standard places no explicit requirements on process. However, the information required by its SQAP makes implicit assumptions regarding process, a process that is more prescriptive than that of IEEE/EIA 12207. This standard assumes a set of meetings, events, and audits through which a certain set of documents are developed and evaluated. By contrast, IEEE/EIA 12207.0-1996 requires a certain body of information without stipulating any particular set of events or documents. Generally, fulfilling the implied process requirements of this standard would go beyond the requirements of IEEE/EIA 12207.0-1996, but would not violate its requirements.

A.2.3 Life cycle data correlation

The information required in an SQAP by this standard and the information required in an SQAP by IEEE/EIA 12207.1-1997 are similar. It is reasonable to expect that a single document could comply with both standards. The main difference is that this standard specifies a particular format, while IEEE/EIA 12207.1-1997 does not. Details are provided in the clause below.

A.3 Document compliance

This clause provides details bearing on a claim that an SQAP complying with this standard would also achieve "document compliance" with the SQAP as prescribed in IEEE/EIA 12207.1-1997. The requirements for document compliance are summarized in a single row of Table 1 of IEEE/EIA 12207.1-1997. That row is reproduced in Table A.1 of this standard.

Table A.1—Summary of requirements for an SQAP excerpted from Table 1 of IEEE/EIA 12207.1-1997

Information item	IEEE/EIA 12207.0-1996 subclause	Kind	IEEE/EIA 12207.1-1997 subclause	References
Software quality assurance plan	6.3.1.3	Plan	6.20	IEEE Std 730-1998 IEEE Std 730.1-1995 ISO 9000-3: 1997 ISO 9001: 1994 ISO 10005: 1995

The requirements for document compliance are discussed in the following subclauses:

- A.3.1 discusses compliance with the information requirements noted in column 2 of Table A.1 as prescribed by 6.3.1.3 of IEEE/EIA 12207.0-1996.
- A.3.2 discusses compliance with the generic content guideline (the "kind" of document) noted in column 3 of Table A.1 as a "plan." The generic content guidelines for a "plan" appear in 5.2 of IEEE/EIA 12207.1-1997.
- A.3.3 discusses compliance with the specific requirements for an SQAP noted in column 4 of Table A.1 as prescribed by 6.20 of IEEE/EIA 12207.1-1997.
- A.3.4 discusses compliance with the life cycle data objectives of Annex H of IEEE/EIA 12207.0-1996 as described in 4.2 of IEEE/EIA 12207.1-1997.

A.3.1 Compliance with information requirements of IEEE/EIA 12207.0-1996

The information requirements for an SQAP are those prescribed by 6.3.1.3 of IEEE/EIA 12207.0-1996. In this case, those requirements are substantively identical to those considered in A.3.3 of this standard.

A.3.2 Compliance with generic content guidelines of IEEE/EIA 12207.1-1997

The generic content guidelines for a “plan” in IEEE/EIA 12207.1-1997 are prescribed by 5.2 of IEEE/EIA 12207.1-1997. A complying plan shall achieve the purpose stated in 5.2.1 and include the information listed in 5.2.2 of IEEE/EIA 12207.1-1997.

The purpose of a plan is:

> IEEE/EIA 12207.1-1997, subclause 5.2.1: Purpose: Define when, how, and by whom specific activities are to be performed, including options and alternatives, as required.

An SQAP complying with this standard would achieve the stated purpose.

Any plan complying with 12207.1-1997 shall satisfy the generic content requirements provided in 5.2.2 of that standard. Table A.2 of this standard lists the generic content items and, where appropriate, references the clause of this standard that requires the same information. It may be concluded that the information required by this standard is sufficient for compliance except as noted in the third column of Table A.2.

A.3.3 Compliance with specific content requirements of IEEE/EIA 12207.1-1997

The specific content requirements for an SQAP in IEEE/EIA 12207.1-1997 are prescribed by 6.20 of IEEE/EIA 12207.1-1997. A complying SQAP shall achieve the purpose stated in 6.20.1 and include the information listed in 6.20.3 of IEEE/EIA 12207.1-1997.

The purpose of the SQAP is:

— IEEE/EIA 12207.1-1997, subclause 6.20.1: Purpose: Define the software quality assurance activities to be performed during the life cycle of the software. Describe the responsibilities and authorities for accomplishing the planned software quality assurance activities. Identify the required coordination of software quality assurance activities with other activities of the project. Identify the tools and the physical and human resources required for the execution of the plan.

An SQAP complying with this standard and meeting the additional requirements of Table A.2 and Table A.3 of this standard would achieve the stated purpose.

An SQAP complying with 12207.1-1997 shall satisfy the specific content requirements provided in 6.20.3 of that standard. Table A.3 of this standard lists the specific content items and, where appropriate, references the clause of this standard that requires the same information. It may be concluded that the information required by this standard is sufficient for compliance except as noted in the third column of Table A.3.

A.3.4 Compliance with life cycle data objectives

In addition to the content requirements, life cycle data shall be managed in accordance with the objectives provided in Annex H of IEEE/EIA 12207.0-1996.

A.3.5 Conclusion

The analysis documented in this annex suggests that any SQAP complying with this standard and the additions specified in Table A.2 and Table A.3 also complies with the requirements of an SQAP in IEEE/EIA 12207.1-1997. In addition, to comply with IEEE/EIA 12207.1-1997, an SQAP shall support the life cycle data objectives of Annex H of IEEE/EIA 12207.0-1996.

Table A.2—Coverage of generic plan requirements by IEEE Std 730-1998

IEEE/EIA 12207.1-1997 generic content	Corresponding clauses of IEEE Std 730-1998	Additions to requirements of IEEE Std 730-1998
a) Date of issue and status	—	Date of issue and status shall be provided.
b) Scope	4.1 Purpose	—
c) Issuing organization	4.3.1 Organization	Issuing organization shall be identified.
d) References	4.2 Reference documents	—
e) Approval authority	4. Software Quality Assurance Plan	—
f) Planned activities and tasks	4.3.2 Tasks	—
g) Macro references (policies or laws that give rise to the need for this plan)	4.2 Referenced documents	Documents motivating the SQAP shall be referenced.
h) Micro references (other plans or task descriptions that elaborate details of this plan)	4.4.2.3 Software Verification and Validation Plan 4.4.2.6 Software Configuration Management Plan 4.5 Standards, practices, conventions, and metrics 4.8 Problem reporting and corrective action 4.9 Tools, techniques, and methodologies	—
i) Schedules	4.3.2 Tasks 4.6 Reviews and audits	The sequencing and relationships of tasks shall be related to a master schedule.
j) Estimates	—	Estimates of resources to be expended in quality assurance tasks shall be provided or referenced.
k) Resources and their allocation	4.3.1 Organization 4.3.3 Responsibilities 4.9 Tools, techniques, and methodologies	—
l) Responsibilities and authority	4.3.1 Organization 4.3.3 Responsibilities	—
m) Risks	4.15 Risk management	—
n) Quality control measures (NOTE—This includes quality control of the SQAP itself.)	4.4.1 Purpose, item b) 4.8 Problem reporting and corrective action	—
o) Cost	—	The costs of SQA activities and resources shall be provided or referenced.
p) Interfaces among parties involved	4.3 Management	—
q) Environment / infrastructure (including safety needs)	4.5 Standards, practices, conventions, and metrics 4.9 Tools, techniques, and methodologies	Safety needs shall be provided or referenced.
r) Training	4.14 Training	—
s) Glossary	—	A glossary of terms used in the SQAP shall be provided.
t) Change procedures and history (NOTE—This includes the change procedures for the SQAP itself.)	4.13 Records collection, maintenance, and retention	Change procedures and history for the SQAP shall be provided or referenced.

Table A.3—Coverage of specific SQAP requirements by IEEE Std 730-1998

IEEE/EIA 12207.1-1997 specific content	Corresponding clauses of IEEE Std 730-1998	Additions to requirements of IEEE Std 730-1998
a) Generic plan information (see Table A.2)	—	—
b) Quality standards...	4.4.1 Purpose, item b) 4.5 Standards, practices, conventions, and metrics	—
... methodologies, procedures, and tools for performing the quality assurance activities (or their references in the organization's official documentation)	4.9 Tools, techniques, and methodologies	—
c) Procedures for contract review and coordination thereof	4.12 Supplier control	Procedures for reviewing and managing the main development contract between the acquirer and supplier should be added to the plan.
d) Procedures for identification, collection, filing, maintenance, and disposition of quality records	4.10 Code control 4.11 Media control 4.13 Records collection, maintenance, and retention 4.6.2.9 Software Configuration Management Plan Review	—
e) Resources ...	4.3.1 Organization 4.3.3 Responsibilities 4.9 Tools, techniques, and methodologies	—
... schedule(s) ...	4.3.2 Tasks 4.6 Reviews and audits	The sequencing and relationships of tasks shall be related to a master schedule.
... and responsibilities for conducting the quality assurance activities	4.3.3 Responsibilities	—
f) Selected activities and tasks from supporting processes such as Verification, Validation, ...	4.4.2.3 Software Verification and Validation Plan	—
... Joint Review, ...	4.6 Reviews and Audits	—
... Audit, and ...	4.4.1 Purpose, item b) 4.6.2.7 In-process audits 4.6.2.5 Functional audit 4.6.2.6 Physical audit	
... Problem Resolution	4.6.1 Purpose, item c) 4.8 Problem reporting and corrective action	—